Analysis Driven Scientific Simulations
Jay Lofstead (gflofst@sandia.gov)

High Level Idea The current model for petascale applications involves dumping large quantities of data for later analysis to generate scientific insights. Some initial attempts at performing online analysis and visualization have been done [5, 4, 3], but they still work assuming the storage of data for later use for checkpoint restarts and further analysis. The single biggest constraint for exascale computing, power, demands minimizing data movement, a high energy cost operation. By shifting the current scientific process from running large-scale batch jobs that generate mountains of data to interactive jobs that only output sufficient data to track the analysis and the provenance of analysis output for reproducibility will enable the scientific discoveries in an exascale environment. By forcing the simulation to be a slave of the analysis and/or visualization framework, scientists will explore the data interactively in an aggregated, condensed form rather than having to post-process data sifting through it for discoveries. The nearly free cost computation will make recalculation of simulation state cheaper and faster than storing and retrieving data. The concept presented by eSiMon [1] is a partial view of how this may work, but the underlying implementation would have to be completely re-architected.

Implications The implications of this shift in thinking affect all levels of the HPC science infrastructure. It will drive new developments in the analysis and visualization layer for interactive control of slave simulation processes, a restructuring of the simulations to be driven by the analysis layer including rewinding and other non-trivial operations, systems software for generating analysis and visualization output with a minimum of data movement, applied mathematics for developing new techniques for the analysis reductions and also for how to rewind or other non-intuitive operations on the simulation, and finally on systems architecture to provide the infrastructure to support the necessary software layers for this all to be a reality. Another impact that does not affect the system design, but must be noted, is how the science community interacts with what was formerly data for validation and confirmation studies not to mention data mining for ancillary results not directly sought by a particular session.

Challenges Addressed **Faults** - Fault recovery for scientific simulations requires an ability to recover some known, good state and continue execution. Execution environments like Charm++ [2], offer a different model where dependencies are tracked rather than requiring storing regular checkpoints. This proposed system would use a similar idea. Instead of a traditional checkpoint model, simulations would only store data at intervals when it is more expensive to recompute the data rather than recompute the data. This energy calculation would be incorporated as part of the running simulation and applied when making decisions on data storage. Any time a fault is detected, relevant, related data can be checked to determine what is missing and must either be recovered or recalculated without interrupting the rest of the running system. Some resources will be shifted from uninterrupted compute areas to balance the recalculation and synchronization.

Power - The key feature of this system is only moving exactly as much data as necessary and favoring moving only derived data to reduce data volumes. Further, the only data moved from the compute area to the visualization and/or analysis system is the data necessary to drive the current display. Any other data should be maintained only sufficiently to support other operations. Recalculation is favored over storage given the relative cost of moving data across the network and any storage devices.

Memory - By moving to an interactive system that strictly limits data movement to only the derived data that is relevant to the current display, the memory hierarchy will be fully used. Data is only stored or transferred according to the data value rather than a perceived notion of the time between failures. The choice of storage location and technology is strictly driven by the energy cost of the data being managed.

Parallelism - The extreme parallelism of an exascale system requires rethinking current computation models to effectively take advantage of these additional compute resources. This model proposes using this extra computation capability, particularly low energy cost computation, as a way to offset data movement.

Maturity Current research has addressed some of the challenges inherent in this proposal. Programming models like Charm++ [2] offer insights in how to organize the simulation software to better support resilience and re-computation. While this programming model is not sufficient to support this proposal, it offers a strong foundation on which to build the additional functionality of tracking the energy cost of data and dynamic storage decisions based on this energy value.

Interactive monitoring systems, such as eSiMon [1] offer a way to explore the output from a simulation as it is being created. Existing visualization systems like ParaView [3] and VisIt [4] offer capabilities to interactively view the progress of the simulation. The challenge with all of these approaches is the need to incorporate storage, at some level, into the workflow. The eSiMon system is built on the set of data files as they are created. In situ approaches, such as what ParaView and VisIt offer

are limited to the integration into the simulation and are driven by the simulation's progress rather than the other way around. The way that these system offer interactive exploration of the data spaces effectively offers the tools application scientists require to generate the scientific insights. The existing gap in functionality is primarily a matter of focus. A shift to promoting these systems as the driver for the simulations will require some developments that drive simulation changes. Primarily, support for selecting features and driving backwards into the simulation past will require the simulation to rewind through either backwards computation or restore from checkpoint and compute forward or backward. This change is primarily a problem for the simulation rather than for the visualization system.

The biggest shift in this model is the change to simulations. Current simulations are built to compute as quickly as possible using a strictly linearly forward model. Instead, this computation will have to be triggered based on input from the user interface. This change is significant and cannot be underestimated. It is absolutely certain this model cannot work for all simulations. For some, there are mathematical dependencies that prevent this from working. For others, there are political or other requirements to preserve data for external analysis. Climate modeling is the primary example of this model. While this data preservation requirement is based on current standards, it is possible to shift this model. Instead, the climate community can publish configurations, starting data, other parameters, and offer virtual machines for execution images affording others to re-execute the runs to verify the output. Alternatively, external users can request access to the user interface and run the simulations themselves to validate the results.

Uniqueness and Novelty This approach is a radical shift in the HPC science paradigm. Existing, petascale systems can also take advantage of these developments, but with less benefit. The key feature of this approach that require the a feature projected to be part of exasacle are the proliferation of low energy computation engines installed as co-processors. These computation engines offer a way to cheaply recreate data rather than using persistent storage.

Applicability Interactive exploration is a traditional learning technique. The current data storage intensive model is an artificial construct based on the capabilities of the machines and evolving existing programming models. With this new computation model, users are offered a framework for manipulating a high-quality analysis and/or visualization environment while using available computing resources to drive this display. Given sufficient network bandwidth, any interactive activity that can benefit from large computation resources will be able to leverage these developments to advance the state of the art in many other fields.

Effort The effort to explore this approach will be spent in a few areas.

1. *Visualization and Analysis Interfaces* - At the user interface level, a relatively small amount of effort is required to shift from the current model to the proposed model. In this case, the extent of the changes is limited to two areas. First, application steering functionality, which has been investigated extensively previously, must be either incorporated or improved. Second, driving applications based on feature appearance or conditions will require new functionality capable of evaluating data sets to determine approximately where to move the computation to continue the display. This work may be extensive and is likely to require more considerable research efforts.

2. *Data Storage and the Memory Hierarchy* - Existing efforts at both data staging and energy efficiency and green computing offer many insights into the the changes necessary to build such a system. The primary gap in the research lies in associating energy with data rather than computation. Initial work into this area has already begun.

3. *Simulation* - Considerable efforts are required for simulations. These efforts are not limited to just the progression of the simulation, but also in the applied math area to determine how, if possible, to run computation backwards within an acceptable accuracy. This backwards computation is not possible for all situations, but will provide a way to manage energy use by moving backwards to a point just before a checkpoint rather than having to go to the previous checkpoint and calculating forward. This approach will offer potentially considerable savings in energy use and time compared with strict forward computation.

4. *Operating System and Other Infrastructure* - Few changes are required to support this model of computation. In general, machine reservations are already covered by existing scheduling systems and existing techniques to communicate between HPC resources, including Teragrid and other systems, already provide the communication technologies required to support this environment.

Acknowledgements

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] Julian Cummings, Jay Lofstead, Karsten Schwan, Alexander Sim, Arie Shoshani, Ciprian Docan, Manish Parashar, Scott Klasky, Norbert Podhorszki, and Roselyne Barreto. Effis: An end-to-end framework for fusion integrated simulation. *Parallel, Distributed, and Network-Based Processing, Euromicro Conference on*, 0:428–434, 2010.

[2] L.V. Kale and Sanjeev Krishnan. CHARM++: A portable concurrent object oriented system based on C++. In *Proceedings of the Conference on Object Oriented Programming Systems, Languages and Applications*, 1993.

[3] K Moreland, D Lepage, D Koller, and G Humphreys. Remote rendering for ultrascale data. *Journal of Physics: Conference Series*, 125(1):012096, 2008.

[4] M. Riedel, T. Eickermann, S. Habbinga, W. Frings, P. Gibbon, D. Mallmann, F. Wolf, A. Streit, T. Lippert, W. Schiffmann, A. Ernst, R. Spurzem, and W.E. Nagel. Computational steering and online visualization of scientific applications on large-scale hpc systems within e-science infrastructures. In *e-Science and Grid Computing, IEEE International Conference on*, pages 483 –490, dec. 2007.

[5] Fang Zheng, Hasan Abbasi, Ciprian Docan, Jay Lofstead, Scott Klasky, Qing Liu, Manish Parashar, Norbert Podhorszki, Karsten Schwan, and Matthew Wolf. PreDatA - preparatory data analytics on Peta-Scale machines. In *In Proceedings of 24th IEEE International Parallel and Distributed Processing Symposium, April, Atlanta, Georgia*, 2010.